# scientific reports

### OPEN



## Inverse binary optimization of convolutional neural network in active learning efficiently designs nanophotonic structures

Jaehyeon Park<sup>1</sup>, Zhihao Xu<sup>2</sup>, Gyeong-Moon Park<sup>3</sup>, Tengfei Luo<sup>2 $\boxtimes$ </sup> & Eungkyu Lee<sup>1 $\boxtimes$ </sup>

Binary optimization using active learning schemes has gained attention for automating the discovery of optimal designs in nanophotonic structures and material configurations. Recently, active learning has utilized factorization machines (FM), which usually are second-order models, as surrogates to approximate the hypervolume of the design space, benefiting from rapid optimization by Ising machines such as quantum annealing (QA). However, due to their second-order nature, FM-based surrogate functions struggle to fully capture the complexity of the hypervolume. In this paper, we introduce an inverse binary optimization (IBO) scheme that optimizes a surrogate function based on a convolutional neural network (CNN) within an active learning framework. The IBO method employs backward error propagation to optimize the input binary vector, minimizing the output value while maintaining fixed parameters in the pre-trained CNN layers. We conduct a benchmarking study of the CNN-based surrogate function within the CNN-IBO framework by optimizing nanophotonic designs (e.g., planar multilayer and stratified grating structure) as a testbed. Our results demonstrate that CNN-IBO achieves optimal designs with fewer actively accumulated training data than FM-QA, indicating its potential as a powerful and efficient method for binary optimization.

**Keywords** Binary optimization, Active learning, Convolutional neural network, Inverse optimization, Nanophotonics

Binary optimization has been applied to a wide range of applications across various fields, from network design in telecommunications<sup>1-3</sup> and vehicle routing in transportation<sup>4-6</sup> to structural optimization in material sciences<sup>7-14</sup>. Various strategies have been used for binary optimization, such as discrete particle swarm optimization (DPSO)<sup>5,15</sup>, genetic algorithm (GA)<sup>16,17</sup>, quadratic unconstrained binary optimization (QUBO)<sup>8,9</sup>, and artificial bee colony (ABC)<sup>18,19</sup>. Recently, an active learning scheme has been proposed to efficiently solve binary optimization tasks to accelerate the discovery of optimal designs in the field of nanophotonics (e.g., planar multilayer (PML) structures, stratified grating systems)<sup>7-9,11-14</sup> and material science (e.g., high-entropy alloys)<sup>10</sup>. This active learning method is iterative and actively accumulates a sparse training dataset consisting of binary vectors and the associated figure-of-merits (FoMs). It uses a second-order factorization machine (FM)<sup>20</sup> to project the design task onto a QUBO model with the training dataset, then leverages quantum annealing (QA)<sup>21</sup> to quickly solve the QUBO problem. For example, Kitai et al.<sup>7</sup> and Kim et al.<sup>8</sup> used the FM-based active learning to design radiative coolers using 40-dimensional binary vectors. They identified an optimal structure with 2000–5000 iterations, which cover ~ 10<sup>-7</sup>% of all the possible binary vector states.

The key to the active learning scheme is that a machine learning model can accurately capture the *N*-dimensional hypervolume of the original binary space near the global and local optima with the actively accumulated sparse training dataset <sup>22,23</sup>. Here, the optimum refers the optimal binary vector yielding the minimum or maximum FoM, or close to it; essentially, it is the point that achieves the target objective. Thus, in each iteration, the formulated machine learning surrogate model should be optimized to identify the best candidate binary vector, which is hopefully near the global optimum. The identified candidate binary vector is then accumulated in the training dataset, which is important for iteratively refining the model and improving its accuracy. However, the FM-based surrogate model in active learning has limitations. The surrogate model (i.e.,

<sup>1</sup>Department of Electronic Engineering, Kyung Hee University, Yongin-si, Gyonggi-do 17104, Republic of Korea. <sup>2</sup>Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA. <sup>3</sup>Department of Artificial Intelligence, Korea University, Seongbuk-gu, Seoul 02841, Republic of Korea. ⊠email: tluo@nd.edu; eleest@khu.ac.kr QUBO model) constructed by the FM includes only linear and quadratic terms, which cannot capture higherorder interactions (e.g., third order, fourth order, and so on) in the *N*-dimensional hypervolume space<sup>20,24</sup>. Consequently, while the QA can identify the best candidate binary vector from the formulated QUBO model, this vector may not be of high quality due to the limitations of the FM in describing the optimization hyperspace.

On the other hand, convolutional neural networks (CNNs), which are traditionally used for image processing tasks<sup>25,26</sup> such as image classification and object detection, can act as surrogate models to approximate complex objective functions. As CNNs are well-suited for feature hierarchies thus efficiently extracting interactions in discrete input variables, they can potentially capture the complexity of the *N*-dimensional hypervolume of the binary space with multiple convolutional layers and activations. However, CNNs are a feed-forward neural network where data flows in one direction from the input layer to the output layer<sup>27</sup>. The optimization of CNN-based surrogate models, therefore, has to leverage external optimization strategies (e.g., DPSO, ABC, or GA), which are often trapped in local minima or maxima as the dimension of the binary space increases<sup>28</sup>. Recently, Partel et al.<sup>29</sup> proposed the so-called 'extremal learning,' which identifies the inputs that minimizes (or maximizes) the output of a neural network in regression problems with frozen parameters (e.g., weights and biases). Extremal learning is, however, designed specifically for continuous variable inputs. Thus, it is not yet known whether the extremal learning is effective for discrete-variable input (i.e., binary input). Additionally, formulating accurate surrogate models based on CNNs may be more computationally expensive than those based on FMs for supervised learning (e.g., CNNs may require more training data than FMs)<sup>30</sup>.

In this study, we propose an inverse binary optimization (IBO) scheme to enhance a CNN-based surrogate function within active learning frameworks and demonstrate its performance in optimizing nanophotonic structures. The IBO employs an extremal learning approach to fix the parameters of a pre-trained CNN while introducing soft variables to identify the optimal input binary vector via backward propagation. Using the CNN-IBO active learning scheme, we optimize a PML structure designed to transmit visible light in the solar spectrum, with binary vectors of length up to N = 40. We benchmark the performance of the CNN-IBO against conventional binary optimization methods, such as DPSO and exhaustive enumeration, and study the influence of parameters like the IBO learning rate. For binary vectors of length  $N \ge 40$ , we apply the CNN-IBO active learning to design various nanophotonic applications, including visible-light filter based on PML structure and asymmetric-light transmitters based on stratified grating structure, and compare the results with the FM-QA method. We find that CNN-IBO can achieve optimal designs with fewer optimization cycles (i.e., a smaller number of training datasets) than FM-QA, highlighting its potential as a promising alternative in binary optimization.

#### Model and method

The binary optimization to design nanophotonic structures is based on the active learning scheme (see Fig. 1a), which consists of the following steps: (i) accumulating the training dataset (X) consisting of binary vectors  $(\vec{x}_i \in [0,1]^N)$  and associated outputs (i.e., FoMs)  $(y_i \in \mathbb{R}_{\geq 0}), X = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_l, y_l)\}$ ; (ii) formulating a surrogate function  $(\hat{y}_i = f(\vec{x}_i))$  with the training dataset; (iii) optimizing the surrogate function to discover a candidate of the optimal binary vector  $(\vec{x}_*)$  that minimizes the surrogate function  $(\vec{x}_* = \operatorname{argmin}_{\vec{x}} f(\vec{x})$  or  $\hat{y}_* = f(\vec{x}_*)$ ); (iv) evaluating the candidate vector by solving Maxwell's equations (e.g., transfer matrix method, TMM). If the candidate vector is already included in the training dataset, a randomly generated binary vector and the associated FoM are added instead; (v) updating the training dataset, and repeating steps (ii), (iii), and (iv). Further details of the active learning scheme can be found in References<sup>8,9</sup>. It is noted that the active learning scheme utilizes both exploration and exploitation, since the training dataset used for formulating a surrogate function contains both discovered binary vectors and randomly generated binary vectors.

The surrogate function is based on a CNN (see Fig. 1b) consisting of the input layer with N nodes, convolutional (Conv) layers (i.e., kernel 1-Conv1 layer, kernel 2-Conv2 layer, etc.), fully-connected (FC) layers (i.e., FC1 layer and FC2 layer), and the output with a single node. In the initial layers, a kernel with the size of five was utilized to extract features from a broader receptive field. In the deeper layers, a kernel with the size of three was employed to increase the depth of network, allowing it to learn more complex and diverse features. For step (iii), an IBO strategy is used (see Fig. 1c) by freezing parameters in the trained CNN, where the IBO can identify  $\vec{x}_*$  yielding  $\hat{y}_*$  by introducing the soft variable vector  $\vec{z} = \{z^1, z^2, ..., z^N\} \in [\mathbb{R}_{\geq 0}]^N$  with gradient descent as:

$$(z^{m})_{t+1} = (x^{m})_{t} - LrA\left(\frac{\partial E(\theta, \vec{z})}{\partial z^{m}}\right),$$

$$(x^{m})_{t+1} = \begin{cases} 1, \ (z^{m})_{t+1} \ge 0.5\\ 0, \ (z^{m})_{t+1} < 0.5 \end{cases},$$
(1)

where  $(z^m)_t$  is the *m*-th element of  $\vec{z}$  at iteration *t* in the gradient descent,  $x^m$  is the *m*-th element of  $\vec{x}_*$ , Lr is the learning rate, A is the function for adaptive moment estimation (i.e., Adam optimizer<sup>31</sup>), and  $E(\theta, \vec{z})$  is the error function which can be defined as:

$$E(\theta, \vec{z}) = a, \text{ where} \hat{y} = f(\vec{z}),$$
 (2)

where  $\theta$  denotes the collection of the weights and biases, which are frozen, in the formulated CNN-based surrogate function. The calculation of the gradient of the error function  $E(\theta, \vec{z})$  with respect to each element  $(z^m)$  of  $\vec{z}$  can be performed by leveraging backward propagation of errors as:



**Fig. 1**. Active learning using convolutional neural networks (CNN) and inverse binary optimization (IBO). (a) An active learning scheme consisting of five steps to automatically identify the optimal nanophotonic structure. (b) A schematic of the CNN framework for pre-training step, depicted in step (ii) of the active learning scheme. The framework consists of two parts: convolutional (Conv) layers and fully-connected (FC) layers. (c) A strategy for IBO of a pre-trained CNN with backward propagation of error, depicted in step (iii) of the active learning scheme. The sky-blue colored nodes indicate fixed parameter values in the Conv and FC layers.

$$\frac{\partial E}{\partial z^m} = \sum_{l}^{L_{1,m}} \frac{\partial E}{\partial \alpha_l^1} \frac{\partial \alpha_l^1}{\partial z^m},$$

$$\frac{\partial E}{\partial \alpha_j^k} = \sum_{l}^{L_{k+1,j}} \frac{\partial E}{\partial \alpha_l^{k+1}} \frac{\partial \alpha_l^{k+1}}{\partial \alpha_j^k}, \quad \text{for } 1 \le k < D$$

$$\frac{\partial E}{\partial \alpha_1^D} = \hat{y} \frac{\partial g_o}{\partial \alpha_1^D},$$
(3)

where  $\alpha_j^k$  is the product-sum-plus-bias for node *j* in layer *k*,  $L_{k,m}$  is a specific set of nodes in layer *k* connected to node *m* in layer *k*-1, and  $g_o$  is the activation function of the output layer (*D*-th layer).

#### Result and discussion

#### Benchmarking study of IBO in active learning scheme

To benchmark the performance of IBO in active learning, we selected the design task of a visible-light filter with a PML structure in the solar spectrum as a testbed. This PML structure is supposed to perfectly transmit visible photons while reflecting ultraviolet and near-infrared photons in the solar spectrum (see Fig. 2a), which is challenging in terms of design due to the large parametric space<sup>8</sup>. This design task was first introduced in the reference<sup>8</sup>, where active learning was used with the FM-QA method. In our study, we aim to overcome the limitations of the FM-based surrogate function in active learning by proposing the CNN-IBO. Given the variety of binary optimization tasks, our choice of the visible-light filter as a testbed is intentional, as it allows us to focus on this specific design task to compare the optimization performance between the FM-QA and the CNN-IBO. The PML has a thickness of 1200 nm, which is divided into an *N* number of pseudo layers. Each pseudo layer is either SiO<sub>2</sub> or TiO<sub>2</sub>, assigned to a binary digit of '1' for SiO<sub>2</sub> and '0' for TiO<sub>2</sub>. This allows a PML structure to be encoded as a binary vector  $\vec{x}_i \in [0, 1]^N$ . The output  $y_i$  for a binary vector can be calculated as

$$y_{i} = \frac{10 \int_{\lambda_{i}}^{\lambda_{f}} \left[ S\left(\lambda\right) \left(T\left(\lambda\right) - T_{I}\left(\lambda\right) \right) \right]^{2} d\lambda}{\int_{\lambda_{i}}^{\lambda_{f}} S\left(\lambda\right)^{2} d\lambda}$$
(4)

where  $T(\lambda)$  and  $T_1(\lambda)$  are, respectively, the transmittance of the designed and target PML, and S ( $\lambda$ ) is the AM 1.5G solar spectrum,  $\lambda$  is the wavelength of the incident photon,  $\lambda_i$  and  $\lambda_f$  are 300 nm and 2500 nm, respectively. In Eq. (4), a smaller output ( $y_i$ ) is preferred for the PML structure (e.g., an ideal visible-light filter has  $y_i = 0$ ).  $T(\lambda)$  is obtained by solving Maxwell's equations with the transfer matrix method. The active learning starts with an initial dataset of 25 (i.e.,  $X = \{(\vec{x}_1, y_1), \dots, (\vec{x}_l, y_l)\}, l=25$ ), and stops at l=2025 (for N=16, 20, and 24) and l=3025 (for N=32, and 40) for the benchmarking study.

The benchmarking results are shown in Fig. 2b-f, where active learning with IBO was performed for N = 16, 20, 24, 32, and 40. The results plot the output  $(\hat{y}_*)$  of the surrogate function for the identified optimal binary vector  $(\vec{x}_*)$  as a function of optimization cycles. For comparison, exhaustive enumeration (EE) and discrete particle swarm optimization (DPSO) were also used to optimize the surrogate function at each optimization cycle, and their results were also plotted. The EE method can identify the global optimal vector of the surrogate function, but it is limited by computational resources. In our study, for N>24, the EE method could not be used due to the computational limitations (e.g., insufficient random-access memory). DPSO, a meta-heuristic method, is known to efficiently solve binary optimization problems. In Fig. 2a, for N=16, 20, and 24, it is seen that the optimized outputs by the IBO method are very close to the global minimum identified by the EE method and are lower than those obtained by DPSO. It is evidenced by the density-of-output distributions (see the subpanels on the right side of each panel in Fig. 2a), showing that the distributions for IBO and EE nearly coincide each, while those for DPSO deviate and show higher values. The deviation becomes more pronounced for N=32and 40. These results demonstrate that IBO can efficiently discover the optimal binary vector close to the global solution of the surrogate function, indicating its methodological compatibility with the CNN-based surrogate function. We note that for N=32 and 40, it could not be verified whether the optimized outputs by the IBO method are close to the global optimal values since EE is too computationally expensive to be performed to find the absolute global minimum.

The results of active learning with IBO are shown in Fig. 2g. This figure tracks the minimum of the output ( $y_i$ ) in the training dataset as a function of the optimization cycles, referred to as the " $y_{min}$  history plot" for each N case. The  $y_{min}$  history plot demonstrates a continuous reduction of  $y_{min}$  as the number of the optimization cycle increases, a characteristics typically observed in the previous studies using active learning<sup>8</sup>. For N = 16, we verified that active learning identified the global optimal binary vector, which is identified by calculating the outputs for all binary vectors using TMM (see Figure S1 in Supporting Information), indicating that the CNN-IBO can find the global optimal point through exploration and exploitation. As the length of the binary vector increases, active learning discovers better structures yielding lower output values. Specifically, the case of N = 40 (or 32) achieved much better optimized structure than that of N = 20 (or 16). These results indicate that active learning with CNN-IBO effectively optimizes the design of the PML structure for the given target function. We also investigated the feature map of the CNN-based surrogate function at N = 24 (see Figure S2 in Supporting Information). The colored visualization of the first convolutional layer reveals the characteristics of binary input variables, while the visualization of the last layer shows a distinguishable distribution that varies with the FOM values. Specifically, the color distributions for input binary vectors associated with lower FOM values are similar



**Fig. 2.** Benchmarking study of CNN-IBO-based active learning. (a) (Top) Schematic model of the planar multilayer (PML) structure for the benchmarking study. (Bottom) The target transmitted irradiance as a function of wavelength. The background grey-colored area shows the AM1.5 G solar spectrum. (**b**–**f**) The optimized output ( $\hat{y}_*$ ) of the CNN-based surrogate function by the IBO method as a function of the optimization cycle at (**b**) N=16, (**c**) N=20, (**d**) N=24, (**e**) N=32, and (**f**) N=40. For comparison, the optimized outputs by the exhaustive enumeration (EE) method and the discrete particle swarm optimization (DPSO) method are also shown. For (**b**–**f**), the right panel shows the projected density of optimized outputs over optimization cycles: (blue) DPSO, (black) EE, (red) IBO. (**g**) The minimum output ( $y_i$ ) in the training dataset as a function of the optimization cycle using CNN-IBO-based active learning at various N=16, 20, 24, 32, and 40.

to one another, indicating that the CNN effectively captures the parametric space relevant to the design task. This suggests that the CNN is capable of generalizing across inputs with similar conditions, an important aspect of its ability to model the underlying design parameters accurately.

On the other hand, the performance of the IBO depends on the learning rate (Lr) of the gradient descent. We investigated the error function (E) as a function of the gradient descent iteration (t in Eq. (1)) with various Lr values: 0.25, 0.5, 1.0, and 2.0 for N=24, as shown in Fig. 3. For each iteration, the error function value at the locally optimized soft variable is depicted as blue dots, while the value for the projected binary space (i.e., local optimal binary vector) is shown as red dots. At learning rates of 0.25 and 0.5, the IBO appears to be trapped in a local optimum (a local optimal binary vector) after t=100, which cannot be overcome by additional iterations. At the learning rate of 1.0, the trapping issue is not observed. The IBO identifies various local optimal binary vectors as well as optimized soft variables, with the error function values continuously decreasing as the number of iteration increases. At a higher learning rate of 2.0, the optimization performance is slightly compromised, resulting in slightly higher optimal error function values compared to those at Lr = 1.0. These results suggest that an appropriate learning rate should be determined to maximize the performance of the IBO.

#### Performance comparison between CNN-IBO and FM-QA methods

Next, we compare the performance of active learning with CNN-IBO to the FM-QA method. For this comparison, we selected two distinct design tasks studied in Refs. [8,9]: optimizing a PML structure with four material bases (SiO<sub>2</sub> = 00, Si<sub>3</sub>N<sub>4</sub> = 01, Al<sub>2</sub>O<sub>3</sub> = 10, TiO<sub>2</sub> = 11) for the given output function of Eq. (2); and identifying a one-dimensional stratified grating (OSG) structure that achieves asymmetric-light transmittance with the output function, given by :  $1 - (T_F - T_B)$ , where  $T_F$  (or  $T_B$ ) is the forward (or backward) transittance at the wavelength of 600 nm. For the PML task, the structure is assumed to consist of twenty-four pseudo layers, each with a thickness of 50 nm, resulting in N=48 ( $4^{24}=2^{48}$ ) (see Fig. 4a). For the OSG task, the grating has a unit cell with a periodic length of 450 nm. The unit cell consists of five thin layers: the middle layer has a thickness of 50 nm, while the other layers each have a thickness of 20 nm. Each layer is discretized into ten rectangular pixels, except the middle layer (see Fig. 4b). A rectangular pixel can be either dielectric (air for the top layer, otherwise



**Fig. 3**. The IBO characteristics depending on the learning rate (*Lr*). (**a**–**d**) The output of the error function as a function of the iteration of the IBO method at various (**a**) Lr=0.25, (**b**) Lr=0.5, (**c**) Lr=1, and (d) Lr=2. The surrogate function used for (**a**) to (**d**) is a pre-trained CNN at N=24 and the 2,205 training dataset. The global minimum is obtained by the EE method.



**Fig. 4.** Performance comparison between the CNN-IBO and FM-QA methods for designing various nanophotonic structures. (**a**) and (**b**) The minimum output in the training dataset as a function of the optimization cycle for designing (**a**) a visible-light filter based on PML and (**b**) an asymmetric-light transmitter based on a one-dimensional stratified grating (OSG). The data for FM-QA in (**a**) is from Ref. [8] and in (**b**) from Ref. [9]. The green-colored dotted line depicts the mean value of the minimum outputs from five different optimizations using CNN-IBO. The green-colored area represents the range of maximum and minimum values, with the minimum value is highlighted by a solid line. Insets: schematics of nanophotonic structures and encoding strategies. The green (or orange) arrow indicates the schematic of the optimized structure using the CNN-IBO (or FM-QA) method. (**c**-**f**) The parity plots of CNN-based and FM-based surrogate functions for cases (**a**) and (**b**). The datasets used for the parity plots are from the results of CNN-IBO in (**a**) and (**b**). (**g**) The transmittance irradiance of the optimized PML structure using FM-QA and CNN-IBO as a function of wavelength. (**h**) The forward and backward transmittance of the optimized asymmetric-light transmitter using FM-QA and CNN-IBO as a function of wavelength.

 $SiO_2$ ) or Ag, and can be assigned '0' for dielectric or '1' for Ag. This configuration of OSG can be encoded as a binary vector with a length of N=40. We noted that the learning rates for these two different design tasks were optimized and set to 'Lr=1.0' for both cases (see Figure S3 in Supporting Information).

We investigated the  $y_{min}$  history plots of the CNN-IBO for the design tasks, as shown in Fig. 4a, b. The history plots of the FM-QA are sourced from Ref.<sup>8</sup> for the PML tasks and Ref.<sup>9</sup> for the OSG tasks. The CNN-IBO method was tested with five trials each for both the PML and OSG tasks, and the history plots display statistical data, including the average, minimum, and maximum values, as a function of the optimization cycles. In both cases, the CNN-IBO effectively minimizes the  $y_{min}$  values, leading to rapid convergence in the history plots as optimization cycles increase. It is evident that the CNN-IBO achieves convergence with fewer optimization cycles compared to the FM-QA across all trials. This consistent performance is observed in both the PML and OSG tasks despite differences in their respective hypervolume spaces. We also investigated the parity plots for both cases, as shown in Fig. 4c-f. For the parity plots, we used the dataset at the optimization cycle of 3000, actively accumulated by the CNN-IBO. In the selected dataset, 80% is used for formulating the CNN (or FM) surrogate function, and 20% is used for validation. It is clear that the CNN-based surrogate function possesses better accuracy than the FM-based surrogate function. The CNN surrogate function for the PML task has an R<sup>2</sup> score of 0.997 for the training set and 0.917 for the validation set, and for the OSG task, it has an R<sup>2</sup> score of 0.997 for the training set and 0.876 for the validation set. These scores are much higher than those of the FM cases (see Fig. 4d, f), and the scores of the validation set are generally considered a good fit. This is also supported by the stable training curves of the validation and training data sets (see Figure S4 in Supporting Information), where early stopping was applied based on the validation loss to prevent overfitting. It is noted that the R<sup>2</sup> score value of the validation data depends on the volume of the CNN, the architecture (e.g., deep neural network, DNN), or the training criterion (e.g., regularization) (see Figure S5 in Supporting Information). With the reduced volume CNN, the R<sup>2</sup> scores of the validation set are 0.9150 for the visible-light filter case and 0.8565 for the asymmetric-light filter case, which are similar but slightly lowered. The DNN and CNN with regularization cases show a noticeable drop in the R<sup>2</sup> scores of the validation set, ranging from 0.82 to 0.87, which are, however, still acceptable values. These results indicate that the CNN-IBO method not only effectively captures the *N*-dimensional hypervolume space but also performs robustly regardless of the specific characteristics of the hypervolume. It also suggests that the CNN can better describe the complexity of the hypervolume of binary space near the optimal point, accounting for the observed optimization performance of the CNN-IBO compared to the FM-QA in the history plots in Fig. 4a, b. We noted that QA is expected to identify the global optimal binary vector (or a local optimal vector very close to the global point) of a given QUBO problem <sup>21,32–36</sup>. Thus, it is reasonable to assume that the performance of the FM-QA method is limited by the accuracy of the FM surrogate model.

#### Conclusion

In conclusion, the CNN-IBO active learning method effectively optimized the design tasks of PML structures by mapping them onto binary optimization problems. For  $16 \le N \le 40$ , the CNN-IBO method demonstrated conventional characteristics of binary optimization using an active learning scheme and efficiently identified optimal PML-based visible-light filter. Within the active learning framework, IBO discovered optimal binary vectors with significantly lower output values compared to the DPSO method for a given CNN-based surrogate function. For higher dimensions ( $N \ge 40$ ), the CNN-IBO method needed fewer optimization cycles than the FM-QA method to identify the optimal designs of visible-light filter and asymmetric-light transmitters. This efficiency is attributed to the capability of the CNN-based surrogate function to capture the complexity of the hypervolume in binary space. However, further investigation is needed to assess the performance of IBO in very high-dimensional binary optimization (e.g.,  $N \sim 100$ ). Additionally, the performance of the IBO should be examined for various kinds of binary optimization problems, and the current strategy of the IBO, which leverages continuous variables and rounds back to binary values, may need to be modified or improved. The IBO method can also be applied to optimize discrete variable or hybrid (e.g., discrete and continuous) variable optimization problems by leveraging deep neural networks. Also, the active learning with the CNN-IBO strategy can be applied to structural design tasks compatible with discrete or binary optimization. For example, lattice optimization<sup>37</sup>, including high-entropy alloys and quaternary semiconductor alloys, can leverage active learning with the CNN-IBO method. Each atom can be mapped with binary digits, and their optimal composition for a good figure-of-merit may be identified. A different architecture, such as a DNN-based surrogate function, can alternatively be used for the CNN-based surrogate model, which can potentially leverage the IBO strategy for optimization.

#### Data availability

Data will be made available on request to Eungkyu Lee (eleest@khu.ac.kr).

Received: 10 February 2025; Accepted: 21 April 2025 Published online: 30 April 2025

#### References

- 1. Zhang, H., Luo, F., Wu, J., He, X. & Li, Y. LightFR: Lightweight federated recommendation with privacy-preserving matrix factorization. ACM Trans. Inf. Syst. 41, 1–28 (2023).
- 2. Alghamdi, M. I. Optimization of load balancing and task scheduling in cloud computing environments using artificial neural networks-based binary particle swarm optimization (BPSO). *Sustainability* **14**, 11982 (2022).
- Douik, A., Sorour, S., Al-Naffouri, T. Y. & Alouini, M.-S. Decoding-delay-controlled completion time reduction in instantly decodable network coding. *IEEE Trans. Veh. Technol.* 66, 2756–2770 (2016).
- Mouhrim, N., El Hilali Alaoui, A. & Boukachour, J. Pareto efficient allocation of an in-motion wireless charging infrastructure for electric vehicles in a multipath network. *Int. J. Sustain. Transp.* 13, 419–432 (2019).
- Mouhrim, N., Alaoui, A. E. H. & Boukachour, J. In 2016 3rd International Conference on Logistics Operations Management (GOL) 1–7 (IEEE).
- Ai, T. J. & Kachitvichyanukul, V. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Comput. Oper. Res.* 36, 1693–1702 (2009).
- 7. Kitai, K. et al. Designing metamaterials with quantum annealing and factorization machines. *Phys. Rev. Res.* 2, 013319 (2020).
- 8. Kim, S. et al. High-performance transparent radiative cooler designed by quantum computing. ACS Energy Lett. 7, 4134-4141 (2022).
- 9. Kim, S. et al. Quantum annealing-aided design of an ultrathin-metamaterial optical diode. Nano Converg. 11, 16 (2024).
- 10. Luo, T., Xu, Z., Shang, W., Kim, S. & Lee, E. QALO: Quantum Annealing-assisted Lattice Optimization (2024).

- 11. Yu, J.-S. et al. Ultrathin Ge-YF3 antireflective coating with 0.5% reflectivity on high-index substrate for long-wavelength infrared cameras. *Nanophotonics* **13**, 4067–4078 (2024).
- 12. Kim, J.-H. et al. Wide-angle deep ultraviolet antireflective multilayers via discrete-to-continuous optimization. *Nanophotonics* **12**, 1913–1921 (2023).
- 13. Kim, S., Jung, S., Bobbitt, A., Lee, E. & Luo, T. Wide-angle spectral filter for energy-saving windows designed by quantum annealing-enhanced active learning. *Cell Rep. Phys. Sci.* 5 (2024).
- 14. Park, G. T. et al. Conformal antireflective multilayers for high-numerical-aperture deep-ultraviolet lenses. Adv. Opt. Mater. 2401040 (2024).
- 15. Khanesar, M. A., Teshnehlab, M. & Shoorehdeli, M. A. In 2007 Mediterranean Conference on Control & Automation 1-6 (IEEE).
- Patle, B., Parhi, D., Jagadeesh, A. & Kashyap, S. K. Matrix-binary codes based genetic algorithm for path planning of mobile robot. Comput. Electr. Eng. 67, 708–728 (2018).
- 17. Kim, S., Lee, S., Choi, J. I. & Cho, H. Binary genetic algorithm for optimal joinpoint detection: application to cancer trend analysis. *Stat. Med.* **40**, 799–822 (2021).
- Kashan, M. H., Nahavandi, N. & Kashan, A. H. DisABC: A new artificial bee colony algorithm for binary optimization. Appl. Soft Comput. 12, 342–352 (2012).
- Santana, C. J. Jr., Macedo, M., Siqueira, H., Gokhale, A. & Bastos-Filho, C. J. A novel binary artificial bee colony algorithm. *Futur. Gener. Comput. Syst.* 98, 180–196 (2019).
- 20. Rendle, S. In 2010 IEEE International Conference on Data Mining 995-1000 (IEEE).
- 21. Pastorello, D. & Blanzieri, E. Quantum annealing learning search for solving QUBO problems. *Quantum Inf. Process.* 18, 303 (2019).
- 22. Settles, B. Active Learning Literature Survey (2009).
- 23. Kumar, P. & Gupta, A. Active learning query strategies for classification, regression, and clustering: A survey. J. Comput. Sci. Technol. 35, 913-945 (2020).
- 24. Rafique, R., Islam, S. R. & Kazi, J. U. Machine learning in the prediction of cancer therapy. Comput. Struct. Biotechnol. J. 19, 4003–4017 (2021).
- Park, G.-M., Yoo, S.-M. & Kim, J.-H. Convolutional neural network with developmental memory for continual learning. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 2691–2705 (2020).
- 26. Shin, H.-C. et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **35**, 1285–1298 (2016).
- 27. Alzubaidi, L. et al. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. J. Big Data 8, 1–74 (2021).
- Kattenborn, T., Leitloff, J., Schiefer, F. & Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. ISPRS J. Photogramm. Remote. Sens. 173, 24–49 (2021).
- Patel, Z. & Rummel, M. Extremal learning: extremizing the output of a neural network in regression problems. Preprint at http:// arxiv.org/abs/2102.03626 (2021).
- He, X. & Chua, T.-S. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval 355–364.
- 31. Kingma, D. P. Adam: A method for stochastic optimization. Preprint at http://arxiv.org/abs/1412.6980 (2014).
- 32. Boixo, S. et al. Evidence for quantum annealing with more than one hundred qubits. Nat. Phys. 10, 218-224 (2014).
- Baldassi, C. & Zecchina, R. Efficiency of quantum versus classical annealing in nonconvex learning problems. Proc. Natl. Acad. Sci. 115, 1457–1462 (2018).
- Mishra, A., Albash, T. & Lidar, D. A. Finite temperature quantum annealing solving exponentially small gap problem with nonmonotonic success probability. *Nat. Commun.* 9, 2917 (2018).
- 35. Boixo, S. et al. Computational multiqubit tunnelling in programmable quantum annealers. Nat. Commun. 7, 10327 (2016).
- 36. Chapuis, G., Djidjev, H., Hahn, G. & Rizk, G. In Proceedings of the Computing Frontiers Conference 63-70.
- 37. Xu, Z., Shang, W., Kim, S., Lee, E. & Luo, T. Quantum annealing-assisted lattice optimization. NPJ Comput. Mater. 11, 4 (2025).

#### Acknowledgements

This research was supported by the Quantum Computing Based on Quantum Advantage Challenge Research (RS-2023-00255442) funded by the Korea government(MSIT) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant (RS-2024-00393808, Efficient design of RF components and systems based on artificial intelligence) funded by the Korea government(MSIT).

#### Author contributions

Jaehyeon Park: Writing-origial draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Conceptualization. Zhihao Xu: Writing-review & editing, Investigation, Gyeong-Moon Park: Writing-review & editing, Investigation, Conceptualization. Tengfei Luo: Writing-origial draft, Validation, Investigation, Conceptualization. Eungkyu Lee: Writing-origial draft, Validation, Software, Resources, Project administration, Funding acquisition, Methodology, Investigation, Conceptualization.

#### **Declarations**

#### **Competing interests**

The authors declare no competing interests.

#### Additional information

**Supplementary Information** The online version contains supplementary material available at https://doi.org/1 0.1038/s41598-025-99570-z.

Correspondence and requests for materials should be addressed to T.L. or E.L.

Reprints and permissions information is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025